

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S3	377	bifurcated near5 process\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:33
S6	4	S5 near5 debug\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:36
S7	130	S5 and @ad < "20030502"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:35
S94	404	bifurcated near5 process\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:38
S95	404	S94	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:33
S96	4	S94 near5 debug\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:34
S97	108	shadow adj process	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:35
S98	75	S97 and @ad < "20030502"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:35

EAST Search History

S99	5	S97 near5 debug\$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:36
S10 0	30102	(work working active current) adj process	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:37
S10 1	6	bifurcated with process\$3 with agent\$1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:40
S10 2	2	S101 and S94	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:40
S10 3	856	(717/124).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:40
S10 4	595	(717/127).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:41
S10 5	592	S103 and @ad < "20030502"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:41
S10 6	458	S104 and @ad < "20030502"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:41

EAST Search History

S10 7	15	S100 and S105	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:42
S10 8	26	S100 and S106	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 08:44
S10 9	91	("20040221271" "20040221269" "5179702" "6195676" "6516460" "5611043" "5764989" "5848274" "6694510" "6698016" "6961928" "20030066054" "20040210876" "5845129" "6542919" "5684598" "5455949" "5471526" "5594904" "5632032" "5724564" "5794047" "5802371" "5970248" "6212612" "6226787" "6212612" "6226787" "6275956" "6378124" "6378125" "6480818" "6587967" "6662310" "6681384" "6804813" "6826749" "6848097" "6851075" "6886081" "6915509" "6978399" "6978444" "6983452" "7120901" "7143392" "20020019976" "20020152228" "20030056199" "20030056200") pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/05 11:53


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

debugging bifurcated program

SEARCH

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **debugging bifurcated program**

 Found **13,965** of **193,448**

Sort results by

relevance

Display results

expanded form

☒ [Save results to a Binder](#)
☒ [Search Tips](#)
☐ [Open results in a new window](#)

 Try an [Advanced Search](#)

 Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [The portable parallel implementation of two novel mathematical biology algorithms in](#)



ZPL

Marios D. Dikaiakos, Daphne Manoussaki, Calvin Lin, Diana E. Woodward

 July 1995 **Proceedings of the 9th international conference on Supercomputing**

Publisher: ACM Press

 Full text available: [pdf\(1.06 MB\)](#)

 Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

2 [Intelligence in scientific computing](#)



Harold Abelson, Michael Eisenberg, Matthew Halfant, Jacob Katzenelson, Elisha Sacks, Gerald J. Sussman, Jack Wisdom, Ken Yip

 May 1989 **Communications of the ACM**, Volume 32 Issue 5

Publisher: ACM Press

 Full text available: [pdf\(1.61 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The authors discuss the development of intelligent techniques appropriate for the automatic preparation, execution, and control of numerical experiments.

3 [Technical reports](#)



SIGACT News Staff

 January 1980 **ACM SIGACT News**, Volume 12 Issue 1

Publisher: ACM Press

 Full text available: [pdf\(5.28 MB\)](#)

 Additional Information: [full citation](#)

4 [Courses: State of the art in interactive ray tracing](#)



Peter Shirley

 July 2006 **Material presented at the ACM SIGGRAPH 2006 conference SIGGRAPH '06**

Publisher: ACM Press

 Full text available: [pdf\(14.08 MB\)](#)

 Additional Information: [full citation](#), [abstract](#)

Recent improvements in computer hardware have allowed ray tracing to be used in some interactive applications. The trends in architecture and expansions of geometric model should increase the use of interactive ray tracing. This course presents recent and often not-yet published work on interactive ray tracing.

5 EUROCAL '85 Abstracts



S. Kamal Abdali

February 1986 **ACM SIGSAM Bulletin**, Volume 20 Issue 1-2

Publisher: ACM Press

Full text available: [pdf\(2.68 MB\)](#) Additional Information: [full citation](#), [abstract](#)

Programming environments have dramatically improved since the time that the major symbolic computation systems such as MACSYMA and REDUCE were developed. The new environments allow direct user interaction with the machine via the mouse, menus, and icons, making obsolete the requirement that the user type instructions to an operating system or language processor to effect desired computations.

6 Assessing software maintainability



Gerald M. Berns

January 1984 **Communications of the ACM**, Volume 27 Issue 1

Publisher: ACM Press

Full text available: [pdf\(781.11 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

How easy is it to maintain a program? To a large extent, that depends on how difficult the program is to understand. A technique to measure program difficulty yields encouraging results.

Keywords: Fortran programs, debugging aids, maintainability measures, program maintainability

7 Parametric shape analysis via 3-valued logic



Mooly Sagiv, Thomas Reps, Reinhard Wilhelm

May 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 3

Publisher: ACM Press

Full text available: [pdf\(1.10 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Shape analysis concerns the problem of determining "shape invariants" for programs that perform destructive updating on dynamically allocated storage. This article presents a parametric framework for shape analysis that can be instantiated in different ways to create different shape-analysis algorithms that provide varying degrees of efficiency and precision. A key innovation of the work is that the stores that can possibly arise during execution are represented (conservatively) using 3-valued l ...

Keywords: 3-valued logic, Abstract interpretation, alias analysis, constraint solving, destructive updating, pointer analysis, shape analysis, static analysis

8 Region-based shape analysis with tracked locations



Brian Hackett, Radu Rugina

January 2005 **ACM SIGPLAN Notices , Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '05**, Volume 40 Issue 1

Publisher: ACM Press

Full text available: [pdf\(205.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper proposes a novel approach to shape analysis: using local reasoning about individual heap locations instead of global reasoning about entire heap abstractions. We

present an inter-procedural shape analysis algorithm for languages with destructive updates. The key feature is a novel memory abstraction that differs from traditional abstractions in two ways. First, we build the shape abstraction and analysis on top of a pointer analysis. Second, we decompose the shape abstraction into a s ...

Keywords: memory leaks, memory management, shape analysis, static error detection

9 Workshop on compositional software architectures: workshop report



May 1998 **ACM SIGSOFT Software Engineering Notes**, Volume 23 Issue 3

Publisher: ACM Press

Full text available: [pdf\(2.91 MB\)](#) Additional Information: [full citation](#), [index terms](#)



10 Formal methods in delay-differential equations



Giampiero Pecelli

June 1991 **Proceedings of the 1991 international symposium on Symbolic and algebraic computation**

Publisher: ACM Press

Full text available: [pdf\(227.10 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)



11 Recent technical reports



December 1979 **ACM SIGACT News**, Volume 10 Issue 3

Publisher: ACM Press

Full text available: [pdf\(1.99 MB\)](#) Additional Information: [full citation](#)



12 Parallel programming with coordination structures



Steven Lucco, Oliver Sharp

January 1991 **Proceedings of the 18th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Publisher: ACM Press

Full text available: [pdf\(1.14 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



13 An extensible approach to imagery of gridded data



Geoffrey Dutton

July 1977 **ACM SIGGRAPH Computer Graphics , Proceedings of the 4th annual conference on Computer graphics and interactive techniques SIGGRAPH '77**, Volume 11 Issue 2

Publisher: ACM Press

Full text available: [pdf\(3.19 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)



A program offering a variety of cartographic techniques for mapping gridded data is described. Dot-distribution maps, several forms of contour maps and screen-toned maps are currently implemented for plotter and vector CRT. The structure and logic of the program is discussed and illustrated. The approach requires only local access to a data grid in a paging environment, allowing large data sets to be manipulated. Maps output may be plotted at any scale, irrespective of the size of the plotting d ...

Keywords: analytic hill-shading, cartography, contour mapping, device independence, dot-distribution mapping, gridded data, halftone imagery, inclined contour mapping,

spatial analysis, spatial gradients, thematic mapping, vector graphics, virtual graphics, virtual memory

14 Recent technical reports



SIGACT News Staff

July 1979 **ACM SIGACT News**, Volume 11 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.77 MB\)](#) Additional Information: [full citation](#)



15 Frontmatter (letters and notices)



Will Tracz

May 2004 **ACM SIGSOFT Software Engineering Notes**, Volume 29 Issue 3

Publisher: ACM Press

Full text available: [pdf\(140.97 KB\)](#) Additional Information: [full citation](#)



16 C in education and software engineering



R. P. Mody

September 1991 **ACM SIGCSE Bulletin**, Volume 23 Issue 3

Publisher: ACM Press

Full text available: [pdf\(867.85 KB\)](#) Additional Information: [full citation](#), [citations](#), [index terms](#)



17 Analysis and debugging: Combining algorithmic debugging and program slicing



Josep Silva, Olaf Chitil

July 2006 **Proceedings of the 8th ACM SIGPLAN symposium on Principles and practice of declarative programming PPDP '06**

Publisher: ACM Press

Full text available: [pdf\(220.75 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Currently, program slicing and algorithmic debugging are two of the most relevant debugging techniques for declarative languages. They help programmers to find bugs in a semiautomatic manner. On the one hand, program slicing is a technique to extract those program fragments that (potentially) affect the values computed at some point of interest. On the other hand, algorithmic debugging is able to locate a bug by automatically generating a series of questions and processing the programmer's answers ...

Keywords: algorithmic debugging, program slicing



18 Debugging object-oriented programs with behavior views



Donglin Liang, Kai Xu

September 2005 **Proceedings of the sixth international symposium on Automated analysis-driven debugging AADEBUG'05**

Publisher: ACM Press

Full text available: [pdf\(764.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A complex software system may perform many program tasks during execution to provide the required functionalities. To detect and localize bugs related to the implementation of these tasks, the software developers must be able to monitor the progress of the tasks during execution and check whether the actions for these tasks have been performed



correctly. This paper presents a debugger to facilitate this monitoring. The debugger introduces a new kind of abstraction, the behavior views, that can b ...

Keywords: debugging language, execution monitoring, secenario-driven

19 A mechanism for efficient debugging of parallel programs



Barton P. Miller, Jong-Deok Choi

November 1988 **ACM SIGPLAN Notices , Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging PADD '88,**
Volume 24 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.22 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper addresses the design and implementation of an integrated debugging system for parallel programs running on shared memory multi-processors (SMMP). We describe the use of flowback analysis to provide information on causal relationships between events in a program's execution without re-executing the program for debugging. We introduce a mechanism called incremental tracing that, by using semantic analyses of the debugged program, makes the flowback ...

20 Event-based debugging of object/action programs



Chu-Chung Lin, Richard J. LeBlanc

November 1988 **ACM SIGPLAN Notices , Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging PADD '88,**
Volume 24 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.25 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)

[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#)

Welcome United States Patent and Trademark Office

☐ Search Session History[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Edit an existing query or compose a new query in the Search Query Display.

Tue, 5 Dec 2006, 9:23:34 AM EST

Search Query Display

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search



Recent Search Queries

- #1 ((bifurcated process)<in>metadata)
- #2 ((bifurcated <sentence> processes)<in>metadata)
- #3 ((bifurcated <sentence> (debug <or> debugging))<in>metadata)
- #4 ((bifurcated <sentence> (debug <or> debugging))<in>metadata)
- #5 ((distributed <sentence> (debugging <or> debug))<in>metadata)



Indexed by
 Inspec®

[Help](#) [Contact Us](#) [Privacy](#)

© Copyright 2006 IEEE